Gradients should stay on path: better estimators of the reverse- and forward KL divergence for normalizing flows

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 045006 https://doi.org/10.1088/2632-2153/ac9455

MACHINE LEARNING
Science and Technology

RECEIVED
17 July 2022

REVISED
15 September 2022

ACCEPTED FOR PUBLICATION
22 September 2022

PUBLISHED
19 October 2022

PAPER

Gradients should stay on path: better estimators of the reverse- and forward KL divergence for normalizing flows

Lorenz Vaitl[1], **Kim A Nicoli**[1,2], **Shinichi Nakajima**[1,2,3] **and Pan Kessel**[1,2,*]

[1] Department of Electrical Engineering & Computer Science, Technische Universität Berlin, Machine Learning Group, Berlin, Germany
[2] BIFOLD—Berlin Institute for the Foundations of Learning and Data, Technische Universität Berlin, Berlin, Germany
[3] RIKEN Center for AIP, 103-0027 Tokyo, Chuo City, Japan
[*] Author to whom any correspondence should be addressed.

E-mail: pan.kessel@tu-berlin.de

Keywords: variational inference, path gradients, mode dropping, normalizing flows, importance sampling

Abstract

We show how to use the path-wise derivative estimator for both the forward reverse Kullback–Leibler divergence for any practically invertible normalizing flow. The resulting path-gradient estimators are straightforward to implement, have lower variance, and lead not only to faster convergence of training but also to better overall approximation results compared to standard total gradient estimators. We also demonstrate that path-gradient training is less susceptible to mode-collapse. In light of our results, we expect that path-gradient estimators will become the new standard method to train normalizing flows for variational inference.

1. Introduction

Many important physical systems can be described by a Boltzmann distribution

$$p(x) = \frac{1}{Z}\exp(-S(x))\,, \qquad (1)$$

where S is the action which is often known in closed form and $Z = \int \mathrm{d}^d x \exp(-S(x))$ denotes the partition function. The partition function is typically intractable, i.e. cannot be calculated as it is a very high-dimensional integral. Nevertheless, well-established Monte-Carlo-Markov-Chain (MCMC) can be used to sample from the target p and allow for the estimation of physical observables. However, MCMC methods become extremely expensive for situations in which subsequent samples of the Markov Chain have large autocorrelation. Such critical slowing down arises for many systems of great physical interest, e.g. for critical phenomena in statistical physics, in the continuum limit of lattice field theories, or for atomistic systems with a large number of local free energy minima in the context of quantum chemistry. As a result, overcoming critical slowing down constitutes one of the most important unsolved problems of modern computational physics.

Recent work [1–9] has proposed to combine generative models with MCMC to overcome critical slowing down. In this context, a particularly promising type of generative model are normalizing flows because they allow for one-shot sampling, provide a normalized density, and can be interpreted as a diffeomorphic field redefinition of the underlying physical degrees of freedom. In this approach, a normalizing flow q is first trained to closely approximate a target density p. Afterward, physical observables can be estimated with the same asymptotic guarantees as for established MCMC methods [1, 8, 10]. This can be achieved by using the flow either for importance sampling or as the proposal density in a Markov Chain. If the target has been learned well, the resulting estimate will not suffer from critical slowing down as samples are drawn (almost) independently. Flow-based methods, therefore, allow us to completely avoid critical slowing down provided that we can train the model q well.

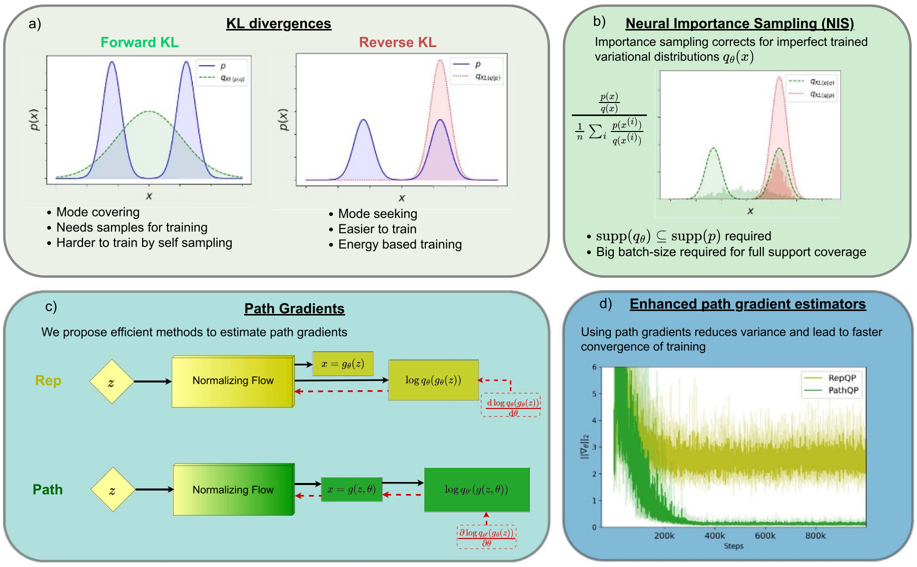
a)
KL divergences
Forward KL
Reverse KL
$p(x)$
x
p
$q_{KL(p|q)}$
$q_{KL(q|p)}$
• Mode covering
• Needs samples for training
• Harder to train by self sampling
• Mode seeking
• Easier to train
• Energy based training
b)
Neural Importance Sampling (NIS)
Importance sampling corrects for imperfect trained variational distributions $q_\theta(x)$
$\frac{\frac{p(x)}{q(x)}}{\frac{1}{n}\sum_i \frac{p(x^{(i)})}{q(x^{(i)})}}$
• $\mathrm{supp}(q_\theta) \subseteq \mathrm{supp}(p)$ required
• Big batch-size required for full support coverage
c)
Path Gradients
We propose efficient methods to estimate path gradients
Rep
Path
z
Normalizing Flow
$x = g_\theta(z)$
$\log q_\theta(g_\theta(z))$
$\frac{\mathrm{d}\log q_\theta(g_\theta(z))}{\mathrm{d}\theta}$
$x = g(z, \theta)$
$\log q_{\theta'}(g(z, \theta))$
$\frac{\partial \log q_{\theta'}(g_\theta(z))}{\partial\theta}$
d)
Enhanced path gradient estimators
Using path gradients reduces variance and lead to faster convergence of training
$||\nabla_\theta||_2$
0
1
2
3
4
5
6
200k
400k
600k
800k
Steps
RepQP
PathQP

Figure 1. Summary of the paper. (a) Forward and reverse KL divergences being respectively *mode covering* and *mode seeking*. The latter is thus more prone to *mode dropping*. (b) Neural importance sampling (NIS) scheme corrects for imperfections of the learned variational distribution when the support of $p(x)$ is covered. (c) We propose to use path gradients and develop an efficient path gradient estimator to improve training efficiency through faster convergence and lower variance. (d) We empirically demonstrate that using path gradients (PathQP) over standard total gradient estimators (RepQP) substantially improves the performances.

Unfortunately, training for normalizing flows represents a major challenge for many physical systems of practical relevance. Current training schemes often minimize the reverse Kullback–Leibler divergence $\mathrm{KL}(q, p)$ from the normalizing flow model q to the target density p and show a drastic deterioration in approximation quality with growing system size or as a critical point is approached. Furthermore, training often results in mode-collapse, i.e. the flow q may assign vanishing probability mass to (at least one) of the modes of the target density p. Mode-collapse must be avoided as it invalidates asymptotic guarantees, i.e. even in the limit of infinitely many samples the estimates of physical observables are biased.

In this work, we propose a plug-and-play modification of the training procedure which alleviates its aforementioned shortcomings and works for any (practically invertible) normalizing flow. Specifically, we propose an algorithm to estimate the path gradient of the reverse KL divergence for normalizing flows. Unlike the conventionally used total gradient, the path gradient only takes into account the implicit dependency on the flow's parameters through reparameterized sampling but is insensitive to any explicit dependency. We demonstrate that the resulting path gradient estimator has lower variance compared to the standard estimator and leads to faster convergence of training as a result. In figure 1, we provide a visual overview of the paper.

Furthermore, we demonstrate that a path-gradient estimator can also be used to minimize the forward Kullback–Leibler divergence $\mathrm{KL}(p, q)$ which is known to be significantly more robust to mode-collapse, see e.g. [11, 12], and therefore is the preferable choice to preserve asymptotic guarantees.

We show in detailed numerical experiments that our path-gradient method leads to superior training results and is able to significantly alleviate mode dropping. We also study these path gradient estimators theoretically by analyzing their statistical properties in various phases of the training process.

1.1. Related works

1.1.1. Path gradients

Broadly speaking, our work builds on and significantly extends Roeder *et al* [13] and Vaitl *et al* [14] which propose path gradient estimators which only work for simple Gaussian variational models or the very restricted subclass of continuous normalizing flows, respectively.

More specifically, Roeder *et al* [13] proposed a path-gradient estimator for the case of a Gaussian variational density and the standard ELBO loss in the context of variational autoencoders (VAE). In Tucker *et al* [15], the authors extended these results to other VAE losses, such as Importance Weighted Autoencoder [16], Reweighted Wake Sleep (RWS) [17], and Jackknife Variational Autoencoder [18]. More specifically, the authors proposed an identity that allows rewriting any REINFORCE-based estimator [19] as a path-wise gradient estimator. Both references empirically demonstrated the superior performance of the path-wise estimators for VAEs. Later work [20–23] extended these results to other VAE loss functions, for example, based on α-divergences, and clarified theoretical aspects of the original references. It is important to stress that all the aforementioned works require a simple variational model, such as a Gaussian.

Our work focuses on normalizing flows which allow modeling complex target distributions of physical systems—in stark contrast to simple Gaussian variational densities. Standard approaches [1, 10] minimize the reverse and forward Kullback–Leibler divergence using the standard total gradient, as opposed to the path gradient, estimator. Extending path-gradient from simple variational densities, such as Gaussians, to normalizing flows is not straightforward as one needs to disentangle the explicit parameter dependence of the model from the implicit one (the latter being related to the parameter dependence of the sampling process). This is because the density involves the (determinant of the) Jacobian of the sampling function thus linking the parameter dependence of the sampling with the one of the density. This is in stark contrast to the case of simple variational densities as considered, for example, in Roeder *et al* [13].

To the best of our knowledge, Agrawal *et al* [24] is the only reference studying path-wise gradient estimators of general invertible normalizing flows. This study is, however, limited to the standard reverse KL as part of a broader ablation for comparatively simple models from the STAN library. In contrast to our contribution, their study does not consider approximating complex distributions of physical systems and path-gradients of forward KL losses. Most importantly, their proposed estimation algorithm has twice the memory costs, severely limiting its suitability for physics applications, as opposed to our proposal.

More recently, Vaitl *et al* [14] introduced an efficient path-gradient estimator for continuous normalizing flows (CNFs). This algorithm for estimating the path gradient—even though efficient—is tailored to CNFs and requires rewriting the gradient computation. On the other hand, our approach is applicable to any practically invertible normalizing flow architecture. In particular, it is also applicable to the case of a CNF. For this specific case, however, it is less efficient than the method proposed in [14].

Therefore, the present work can be thought of as a generalization of [14] any practically invertible normalizing flow architectures. A further notable novelty of our work is that we discuss the estimation of the path gradient of the forward KL for normalizing flows and theoretically analyze the variance properties of the various estimators.

1.1.2. Other variance reduction methods

Most Monte Carlo gradient estimators belong to one of two classes, i.e. reparameterized [25] and REINFORCE-type estimators [19]. The reparameterization trick provides a simple and efficient way of computing low-variance gradients but is however limited to continuous random variables as well as a restricted class of base distributions. It was however generalized in various ways: Figurnov *et al* [26] enhance the applicability of the reparameterization trick to a wider class of distributions using implicit differentiation. Interestingly, this approach was recently also generalized to normalizing flows [27]. Ruiz *et al* [28] relaxes the underlying assumptions of the reparameterization trick by defining invertible transformations such that the base distribution is only weakly dependent on the variational parameters. The reparameterization trick was also generalized by Jankowiak and Obermeyer [29] by harnessing its correspondence to optimal transport. In Wan *et al* [30], the authors extend the reparametrization trick to f-divergences. Naesseth *et al* [31] introduce a reparameterization trick through accept–reject sampling. In principle, these approaches can be combined with path gradients by deriving a path gradient of the corresponding reduced variance objectives. This represents an interesting line of future research.

The REINFORCE-type gradient estimators are more general than the reparameterization trick as only the score of the variational distribution is required and are also applicable to discrete random variables. However, it is well-known to generically have higher variance. Therefore, several variance-reduction methods based on control variates have been proposed. The simplest and most widely-used approach [32] subtracts the expected weighting term. However, more sophisticated methods to choose the coefficient of the control variate have been proposed. Specifically, Richter *et al* [33] building on earlier work by Salimans and Knowles [34] has shown that one can (up to a correction which depends on the value of the reverse KL divergence) obtain the optimal coefficient by using a modified loss function. Even though the path gradients may be thought of as reparametrized gradient estimators with control variates, the aforementioned control variates are based on certain means over the mini-batch. This is not the case with path gradients, that subtract part of the gradient *per sample*.

Self-normalized importance sampling, as applied in our forward-KL estimators, has been shown to reduce the variance of estimators [35–37], other schemes for reducing variance using importance sampling include Adaptive Importance Sampling [38] and Annealed Importance Sampling [39, 40]. It would be interesting to generalize our path gradient estimator of the forward KL along similar lines as part of future research.

1.1.3. Other methods to mitigate mode collapse

Noé *et al* [1] propose to use a convex combination of both the forward and reverse KL divergence to avoid mode-collapse. This approach, however (at the very least) requires a single sample from each mode[4] which can be difficult for situations in which the (local) ground state structure is unknown, such as lattice gauge theories. Laszkiewicz *et al* [41] and Jaini *et al* [42] propose methods to improve learning the tail behavior of a normalizing flow—a property that is crucial in avoiding mode collapse during training. Another well-known strategy to alleviate mode-collapse [43] is to supplement reverse KL training with variance maximization of certain observables. In many applications, it is however not obvious which observable should be used.

Dhaka *et al* [36] investigate mode dropping for an array of different divergences, the effect of dimensionality as well as the effect of self-normalized importance sampling. It is well-known that the one-parameter family of α-divergences includes both the forward and reverse KL divergence. However, other members of this family can also be used as an objective function. As an example $\alpha = 2$, i.e. the χ^2-divergence, has been applied to normalizing flows [10, 36, 40], as well as to Gaussian distributions with path gradients [23]. However, it seems to perform inferior to the standard KL divergence without any additional algorithmic tricks [10, 23, 36]. Recently, however, Midgley *et al* [40] use this choice of α-divergence combined with AIS and a replay buffer for quantum chemistry problems. Applying the path gradient estimators in conjunction with these additional algorithmic measures is not straightforward but an interesting future direction.

1.2. Sampling with normalizing flows

A normalizing flow is a bijective map $g_\theta : \mathcal{Z} \to \mathcal{X}$ from a base space $\mathcal{Z} \subset \mathbb{R}^d$ to a target space $\mathcal{X} \subset \mathbb{R}^d$. The base space $\mathcal{Z}$ is equipped with a simple probability density q_Z. The bijection g_θ then induces a probability density q_θ on the target space $\mathcal{X}$ by

$$q_\theta(x) = q_Z(g_\theta^{-1}(x)) \left| \frac{\partial g_\theta^{-1}(x)}{\partial x} \right| , \qquad (2)$$

where $\left| \frac{\partial g_\theta^{-1}(x)}{\partial x} \right|$ denotes the absolute value of the determinant of the Jacobian of the inverse flow g_θ^{-1}.

The flow is trained to closely approximate the target density p. As we will explain in detail in the next section, this can be done even for an intractable partition function.

After training, we can use the flow q_θ to sample from the target density p with asymptotic guarantees. To this end, one uses NIS [1, 8, 10] to estimate the expectation value of some observable $\mathcal{Q}$ with respect to the target density p by

$$\mathbb{E}_{x\sim p}\left[\mathcal{Q}(x)\right] = \mathbb{E}_{x\sim q_\theta}\left[w(x)\,\mathcal{Q}(x)\right] \approx \frac{1}{N}\sum_{i=1}^{N} \hat{w}_i\,\mathcal{Q}(x_i)\,, \qquad \text{with } \; x_i \sim q_\theta\,, \qquad (3)$$

where we have defined the normalized importance weight

$$w(x) = \frac{p(x)}{q(x)}\,,$$

and its estimator $\hat{w}$ as

$$\hat{w}_i = \hat{w}(x_i) = \frac{1}{\hat{Z}} \frac{\exp(-S(x_i))}{q_\theta(x_i)}\,, \qquad (4)$$

which uses the estimator $\hat{Z}$ of the partition function[5]

$$Z = \int \mathrm{d}^d x\, q_\theta(x) \frac{\exp(-S(x))}{q_\theta(x)} \;\approx\; \hat{Z} = \frac{1}{N}\sum_{i=1}^{N} \frac{\exp(-S(x_i))}{q_\theta(x)}\,, x_i \sim q_\theta\,.$$

The variance of the estimator (3) is given by

$$\sigma^2 = \mathrm{Var}(\mathcal{Q}) \frac{1}{N\,\mathrm{ESS}} + o_p(N^{-1})$$

[4] This assumption may be relaxed to all modes for which the considered observable of interest has sufficiently large support, such as in certain applications of Quantum Chemistry.

[5] We suppress the dependence of the estimator $\hat{Z}$ on the number of samples N to alleviate notation.

with the effective sampling size

$$\mathrm{ESS} = \frac{1}{\mathbb{E}_q[w(x)^2]} \in [0,1]\,. \tag{5}$$

The effective sampling size (ESS) is one for a perfectly trained sampler while very low for a poorly trained one. The effective sampling size therefore provides a natural metric to quantify the quality of a sampler.

If the model density q has larger or equal support than the target density, i.e.

$$\mathrm{supp}(p) \subseteq \mathrm{supp}(q_\theta)\,, \tag{6}$$

normalized importance sampling (3) provides a statistically consistent estimator of the expectation value $\mathbb{E}_p[\mathcal{O}]$ and thus has the same asymptotic guarantees as well-established Monte-Carlo-Markov-Chain (MCMC) methods. We refer to Nicoli *et al* [8] for a detailed proof. In contrast to popular MCMC algorithms, this method uses independent and identically distributed (iid) samples from the flow. Flow-based sampling may therefore have considerable advantages over established Markov chain techniques for situations in which MCMC suffers from large autocorrelation or has problems overcoming large barriers in the action landscape.

We also mention in passing that flow-based sampling can be combined with MCMC by using the flow as the proposal density of the Markov chain [1, 3, 8]. The resulting algorithm is called Neural MCMC. The proposal is drawn independently and therefore does not depend on the previous element of the chain. This is in stark contrast to conventional MCMC algorithms which rely on a (typically small) random modification of the previous configuration to create a proposal. Crucially, Neural MCMC also only comes with asymptotic guarantees if the flow has larger support than the target, i.e. provided that (6) holds.

As we will discuss, current training approaches often lead to a violation of the larger support requirement (6) due to mode-collapse. One of the central motivations of the present work is to propose modifications to the training procedure to alleviate this effect.

1.3. Training with reverse KL

The flow can be trained by minimizing the reverse KL divergence

$$\mathrm{KL}(q_\theta, p) = \mathbb{E}_{x\sim q_\theta}\left[S(x) + \log q_\theta(x)\right] + \mathrm{const.}\,, \tag{7}$$

where the last summand denotes terms independent of the parameters θ of the flow q_θ. As a result, this term will have no contribution to the gradient of the loss function. This gradient can be rewritten using the reparametrization trick, i.e. $\mathbb{E}_{x\sim q_\theta}[f(x)] = \mathbb{E}_{z\sim q_z}[f(g_\theta(z))]$, and is then given by

$$\frac{d}{d\theta}\mathrm{KL}(q_\theta, p) = \mathbb{E}_{z\sim q_Z}\left[\frac{d}{d\theta}S(g_\theta(z)) + \frac{d}{d\theta}\log q_\theta(g_\theta(z))\right]\,. \tag{8}$$

It is straightforward to obtain a Monte-Carlo estimator of this gradient,

$$\frac{d}{d\theta}\mathrm{KL}(q_\theta, p) \approx \mathcal{G}_{\mathrm{RepQP}}\,, \tag{9}$$

by drawing samples from the base density q_Z, reparametrizing and calculating the gradients of the action and of the log-probability by backpropagation, i.e.

$$\mathcal{G}_{\mathrm{RepQP}} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{d}{d\theta}S(g_\theta(z_i)) + \frac{d}{d\theta}\log q_\theta(g_\theta(z_i))\right), z_i \sim q_Z\,. \tag{10}$$

We will refer to $\mathcal{G}_{\mathrm{RepQP}}$ as the *reparameterized qp-estimator (RepQP)*. Currently, this estimator is the most widely used.

However, the RepQP estimator can often be suboptimal and we propose to instead use the *path-gradient estimator*. For this, it is convenient to define the path-gradient of an arbitrary function $f(g_\theta(z),\theta)$ by

$$\blacktriangledown_\theta f(g_\theta(z),\theta) = \frac{\partial f(g_\theta(z),\theta)}{\partial g_\theta(z)}\frac{\partial g_\theta(z)}{\partial\theta}\,,$$

which implies that its total derivative can be written as

$$\frac{d}{d\theta}f(g_\theta(z),\theta) = \blacktriangledown_\theta f(g_\theta(z),\theta) + \frac{\partial}{\partial\theta}f(x,\theta)\bigg|_{x=g_\theta(z)}\,, \tag{11}$$

i.e. the path derivative only takes into account the implicit dependency on θ through the flow g_θ and is insensitive to any explicit dependency. Using this definition, we can rewrite the gradient (8) of the KL-divergence as

$$\frac{d}{d\theta}\mathrm{KL}(q_\theta, p) = \mathbb{E}_{z\sim q_Z}\left[\blacktriangledown_\theta S(g_\theta(z)) + \blacktriangledown_\theta \log q_\theta(g_\theta(z))\right] + \mathbb{E}_{z\sim q_Z}\left[\frac{\partial}{\partial\theta}\log q_\theta(x)\Big|_{x=g_\theta(z)}\right],$$

where we have used that the path and total gradient lead to the same result for the action as

$$\frac{d}{d\theta}S(g_\theta(z)) = \frac{\partial S(g_\theta(z))}{\partial g_\theta(z)}\frac{\partial g_\theta(z)}{\partial\theta} = \blacktriangledown_\theta S(g_\theta(z))\,. \tag{12}$$

By applying the reparameterization trick again, it is easy to see that the last *score term* vanishes

$$\mathbb{E}_{z\sim q_Z}\left[\frac{\partial}{\partial\theta}\log q_\theta(x)\Big|_{x=g_\theta(z)}\right] = \mathbb{E}_{x\sim q_\theta}\left[\frac{\partial}{\partial\theta}\log q_\theta(x)\right] = \frac{\partial}{\partial\theta}\int \mathrm{d}^d x\, q_\theta(x) = 0\,. \tag{13}$$

By explicitly excluding the vanishing score term from the gradient of the KL-divergence, we obtain the *path-gradient qp-estimator (PathQP)*

$$\frac{d}{d\theta}\mathrm{KL}(q_\theta, p) \approx \mathcal{G}_{\mathrm{PathQP}}\,, \tag{14}$$

which is given by

$$\mathcal{G}_{\mathrm{PathQP}} = \frac{1}{N}\sum_{i=1}^{N}\left(\blacktriangledown_\theta S(g_\theta(z_i)) + \blacktriangledown_\theta \log q_\theta(g_\theta(z_i))\right)\; z_i \sim q_Z\,. \tag{15}$$

Both the path-gradient and the reparameterized *qp*-estimator are unbiased estimators of the gradient of the reverse KL divergence. However, their variances are generically different. This effect is particularly pronounced if the variational distribution perfectly approximates the target, i.e.

$$\forall x \in \mathcal{X} : q_\theta(x) = p(x)\,.$$

For such a perfect approximation, the path-gradient estimator vanishes identically:

$$\begin{aligned}\mathcal{G}_{\mathrm{PathQP}} &= \frac{1}{N}\sum_{i=1}^{N}\left(\blacktriangledown_\theta S(g_\theta(z_i)) + \blacktriangledown_\theta \log q_\theta(g_\theta(z_i))\right)\\ &= -\frac{1}{N}\sum_{i=1}^{N}\blacktriangledown_\theta \log\underbrace{\left(\frac{p(g_\theta(z_i))}{q_\theta(g_\theta(z_i))}\right)}_{=1} = 0\,,\end{aligned} \tag{16}$$

where we have used that $-\blacktriangledown_\theta \log p(g_\theta(z)) = \blacktriangledown_\theta S(g_\theta(z))$. As a result, the variance of the path gradient estimator $\mathcal{G}_{\mathrm{PathQP}}$ vanishes in this limit. This is in contrast to the reparameterized estimator which can be rewritten as

$$\mathcal{G}_{\mathrm{RepQP}} = \mathcal{G}_{\mathrm{PathQP}} + \mathcal{G}_{\mathrm{Score}} \tag{17}$$

where we have defined

$$\mathcal{G}_{\mathrm{Score}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial}{\partial\theta}\log q_\theta(g_\theta(z_i)), \quad z_i \sim q_Z\,, \tag{18}$$

whose variance is given by $\mathrm{Var}[\mathcal{G}_{\mathrm{Score}}] = \frac{\mathcal{I}(\theta)}{N}$ where we have defined the Fisher information

$$\mathcal{I}(\theta) = \mathbb{E}_{x\sim q_\theta}\left[\frac{\partial}{\partial\theta}\log q_\theta(x)\frac{\partial}{\partial\theta}\log q_\theta(x)\right] \tag{19}$$

of the variational distribution. As a result, the reparameterized gradient estimator $\mathcal{G}_{\mathrm{RepQP}}$ has generically non-vanishing variance even if the variational distribution perfectly approximates the target distribution. By continuity, we may expect that the variance of the reparameterized estimator is substantially larger than the path-gradient estimator in the final phase of training and that using the path-gradient estimator will lead to a better convergence of training as a result. We will indeed demonstrate this in the numerical experiments.

2. Training with forward KL

As discussed in section 1.2, neural importance sampling (NIS) and neural Markov-Chains (NMCMC) require the flow q_θ to have larger support than the target density p in order to be statistically consistent. Training with the reverse KL divergence is therefore problematic as it can lead to mode-collapse. Furthermore, in practice, it may also be the case that the flow assigns only an infinitesimal probability mass to modes of the target density p. In this case, importance sampling will not lead to reasonable results for any feasible number of samples (although strictly speaking, it is still statistically consistent in the limit of an infinite number of samples).

For this reason, it is preferable to train the flow by minimizing the forward KL divergence,

$$\mathrm{KL}(p, q_\theta) = \mathbb{E}_{x\sim p}\left[\log\left(\frac{p(x)}{q_\theta(x)}\right)\right], \tag{20}$$

as it heavily penalizes mode-collapse. In appendix A, we show that the gradient of the forward KL can be rewritten in terms of the path-gradient:

$$\frac{d}{d\theta}\mathrm{KL}(p, q_\theta) = -\mathbb{E}_{z\sim q_Z}\left[\blacktriangledown_\theta \frac{p(g_\theta(z))}{q_\theta(g_\theta(z))}\right] \tag{21}$$

In the next section, we discuss estimators of this forward path-gradient.

2.1. Estimators for the forward KL path-gradient

There are two possibilities for obtaining an estimator for the forward KL path gradient (21). To see why we recall that the exact normalized weight is defined as

$$w(x) = \frac{1}{Z}\tilde{w}(x), \tag{22}$$

where $\tilde{w} = \frac{\exp(-S(x))}{q_\theta(x)}$ is the unnormalized importance weight. Using these definitions, we can rewrite the path-gradient (21) as

$$\frac{d}{d\theta}\mathrm{KL}(p, q_\theta) = -\mathbb{E}_{z\sim q_Z}\left[\blacktriangledown_\theta w(g_\theta(z))\right] \approx -\frac{1}{N}\sum_{i=1}^{N}\blacktriangledown_\theta \frac{\tilde{w}(g_\theta(z_i))}{Z}.$$

Since the partition function Z is intractable, we need to estimate it with samples from q by

$$Z \approx \hat{Z} = \frac{1}{N}\sum_{j=1}^{N}\tilde{w}(g_\theta(z_i)) z_i \sim q_Z. \tag{23}$$

There are now two ways of obtaining path-gradient estimators:

- We can either pull Z through the path-derivative and then estimate

$$\frac{d}{d\theta}\mathrm{KL}(p, q_\theta) \approx \frac{-1}{\hat{Z}}\frac{1}{N}\sum_{i=1}^{N}\blacktriangledown_\theta \tilde{w}(g_\theta(z_i)) = -\sum_{i=1}^{N}\frac{\tilde{w}_i}{\sum_{j=1}^{N}\tilde{w}_j}\blacktriangledown_\theta \log(\tilde{w}_i) \tag{24}$$

  We will refer to this estimator as the *path-gradient pq-estimator (PathPQ)*.
- Alternatively, we can let the path derivative act on the estimated partition function

$$\begin{aligned}\frac{d}{d\theta}\mathrm{KL}(p, q_\theta) &\approx -\frac{1}{N}\sum_{i=1}^{N}\blacktriangledown_\theta \frac{\tilde{w}(g_\theta(z_i))}{\hat{Z}}\\ &= -\sum_{i=1}^{N}\left(\frac{\tilde{w}_i}{\sum_{j=1}^{N}\tilde{w}_j} - \frac{\tilde{w}_i^2}{(\sum_{j=1}^{N}\tilde{w}_j)^2}\right)\blacktriangledown_\theta \log(\tilde{w}_i)\end{aligned} \tag{25}$$

  We will refer to this estimator as *Z path-gradient pq-estimator (ZPathPQ)*.

As a baseline, we will also consider an estimator which is not based on path-gradients:

- To this end, we estimate (37) by

$$\frac{d}{d\theta}\mathrm{KL}(p,q_\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{\tilde{w}_i}{\hat{Z}}\frac{\partial}{\partial\theta}\log\left(\tilde{w}_i\right), \tag{26}$$

and refer to it as *reinforce pq estimator (ReinfPQ)*.

We note that, to the best of our knowledge, the PathPQ and ZPathPQ estimators were first used in the context of RWS training of VAE by Finke and Thiery [20] and Tucker *et al* [15] for simple Gaussian variational distributions, respectively.

Both the PathPQ and the ZPathPQ estimator have vanishing variance in the limit of perfect approximation, i.e. $q_\theta \equiv p$. This immediately follows by a completely analogous argument as for the reverse KL case, see (16). Similarly, the variance of the ReinfPQ estimator is proportional to the Fisher information of the variational distribution q_θ in this limit and is thus generically non-vanishing. We refer to appendix B for a proof.

One may therefore again expect that the path gradient estimators have lower variance than the reinforce baseline in the final phase of training and will thus lead to better convergence of training. We will verify this in the numerical experiments.

2.2. Theoretical analysis of path gradient estimators

2.2.1. Initial training phase

estimating the forward KL divergence by reweighting can be challenging in the initial phase of training as the density of the flow q_θ and the target p will have a small overlap. In order to analyze this initial training regime theoretically, we will assume, without loss of generality, that all samples $\{x_i\}_{i=1}^{N-1}$ but one sample x_N drawn from the flow q_θ will be in regions of the sampling space for which the target density is very small and flat, i.e.

$$p(x_i) = \mathcal{O}(\epsilon), \quad \nabla p(x_i) = \mathcal{O}(\epsilon), \qquad\qquad q_\theta(x_i) = \mathcal{O}(1), \quad \nabla q_\theta(x_i) = \mathcal{O}(1), \tag{27}$$

for small $\varepsilon > 0$ and $i \in \{1,\ldots,N-1\}$[6]. Since most interesting densities in physics have an exponential fall-off around their modes, this is a reasonable assumption in practice. Under the further mild assumption that $\frac{\partial x}{\partial\theta}$ does not diverge, this implies

$$\blacktriangledown_\theta w_i = \mathcal{O}(\epsilon)\,, w_i = \mathcal{O}(\epsilon)\,. \tag{28}$$

We then show in the appendix D that the PathPQ estimator (24) is

$$\sum_{i=1}^{N}\frac{\tilde{w}_i}{\sum_{j=1}^{N}\tilde{w}_j}\blacktriangledown_\theta\log(\tilde{w}_i) = \frac{\blacktriangledown_\theta\tilde{w}_N}{\tilde{w}_N} + \mathcal{O}(\epsilon) \tag{29}$$

while the ZPathPQ estimator (25) has no order one contribution

$$\sum_{i=1}^{N}\left(\frac{\tilde{w}_i}{\sum_{j=1}^{N}\tilde{w}_j} - \frac{\tilde{w}_i^2}{(\sum_{j=1}^{N}\tilde{w}_j)^2}\right)\blacktriangledown_\theta\log(\tilde{w}_i) = \mathcal{O}(\epsilon)\,. \tag{30}$$

The ZPathPQ estimator (25) can thus be expected to struggle in the initial training phase. We empirically confirm this in section 3.

2.2.2. Asymptotic training phase

If the flow density q_θ already approximates the target p relatively well, such that the normalized importance weight variance is small, we can use the delta method to calculate the variance and bias of the estimators. We show in the appendix C that both the PathPQ and ZPathPQ estimators have the same variance and comparable bias to leading order in the number of samples N. Thus both estimators can be expected to lead to a similar performance in this training regime.

[6] While not explicit in the notation, we assume that $q(x_i)$ is not smaller than an order one number. This assumption is reasonable as x_i is a sample of the flow $q(x_i)$.

Algorithm 1: Path gradient $\blacktriangledown_\theta \log q_\theta(g_\theta(z))$.

```
Input: base sample $z \sim q_Z$
  $x' \leftarrow \text{stop\_gradient}(g_\theta(z))$                                      # forward pass of $z$ through the flow without gradients
  $q_\theta(x') \leftarrow q_Z(g_\theta^{-1}(x')) \left|\frac{\partial g_\theta^{-1}(x')}{\partial x'}\right|$      # reverse pass to calculate density
  $G \leftarrow \frac{\partial \log(q_\theta(x'))}{\partial x'}$                                  # compute gradient with respect to $x'$
  $x \leftarrow g_\theta(z)$                                                    # standard forward pass
  return $\frac{d}{d\theta}\left(\text{stop\_gradient}(G)^T x\right)$                          # for path-gradient, contract $\frac{\partial x}{\partial \theta}$ with $\partial \log q_\theta(x)/\partial x$
```

2.3. Implementation of estimators

In this section, we will discuss the practical implementation of the path gradient for normalizing flows. See appendix F for an implementation.

For both forward estimators (24) and (25) as well as the reverse estimator, we need to calculate

$$\blacktriangledown_\theta \log(\tilde{w}) = \blacktriangledown_\theta \log(q_\theta(g_\theta(z))) + \blacktriangledown_\theta S(g_\theta(z)) \,.$$

The second term can trivially be obtained by automatic differentiation because the total derivative leads to the same result as the path gradient, i.e. $\frac{d}{d\theta}S(g_\theta(z)) = \blacktriangledown_\theta S(g_\theta(z))$, see (12).

The first term however does not have this property and therefore requires more care. For variational inference, the log density of a normalizing flow is typically computed along with the forward pass through the flow $x = g_\theta(z)$ which produces the sample x. This makes it challenging to calculate the path-derivative $\frac{\partial \log q_\theta(x)}{\partial x}$ by standard reverse-mode automatic differentiation because the sample x is the output as opposed to the input of the forward pass.

We overcome this challenge, by proposing algorithm 1 which estimates the path-gradient with the same memory footprint as needed for the conventional gradient at the cost of roughly two forward passes through the flow. In practice, a low memory footprint is crucial because invertible architectures tend to have large memory requirements which severely limits the possible batch sizes. Large batch sizes are however essential for successful training of flows by self-sampling. This is because training starts from a randomly initialized flow. If the batch size is not large enough, the probability of probing regions of the sampling space with significant probability mass tends to be low and the flow will not be able to learn to approximate the target p as it has not sampled these relevant regions. This effect is particularly pronounced as the system size increases as the action S is an extensive quantity and the target density p, therefore, becomes increasingly concentrated around its local minima.

While our algorithm has approximately double runtime per gradient update compared to the standard total gradient estimator, it will be shown empirically that, for practically invertible normalizing flows, it leads to faster convergence of training overall as may be expected due to favorable variance properties discussed in section 1.3. By practically invertible, we mean that the normalizing flow can be evaluated in the reverse direction for roughly the same cost as in the forward direction. Normalizing flows which are not of this type, i.e. their forward pass is significantly cheaper than their reverse pass, will entail a larger overhead per gradient update and thus may not benefit from faster overall convergence.

3. Numerical experiments

3.1. Quantum mechanical particle in double-well

In order to evaluate the performance of training with path-gradients, we consider a quantum mechanical particle in a double well potential which is a prototypical example for a two-moded distribution in quantum mechanics.

In quantum mechanics, the position of the particle at euclidean time t, and thus its path $x(t)$, is a random variable. For a discretized path $x = (x_0, x_1, \ldots, x_{T-1})$, the corresponding density is given by

$$p(x) = \frac{1}{Z}\exp(-S(x)) \tag{31}$$

where $Z = \int d^T x \exp(-S(x))$ is the partition function and the action is given by

$$S(x) = a\sum_{t=0}^{T-1}\left(\frac{m_0}{2}\left(x_{t+1} - x_t\right)^2 + V(x_t)\right) \tag{32}$$

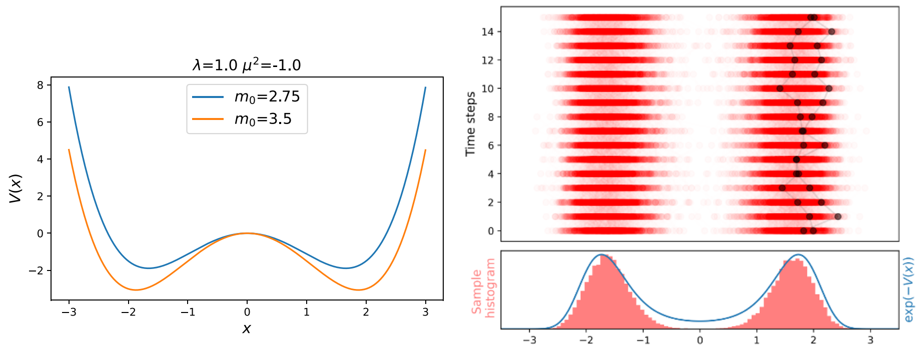

Figure 2. Left: double-well potential for various choices of the mass m_0. Right: visualization of samples from a trained model.

with periodic boundary conditions for x_i and a denoting the lattice spacing. The double-well potential V is defined by

$$V(x) = \frac{m_0\mu^2}{2}x^2 + \frac{\lambda}{4}x^4 \,, \tag{33}$$

where the mass parameters m_0 and μ^2, as well as the coupling λ control the shape of the potential, see the left part of figure 2.

3.2. Forward and reverse effective sampling size

In our numerical experiments, we evaluate the degree to which the model q_θ approximates the target density p. As explained in section 1.2, the effective sampling size (5) is a natural metric to quantify this.

We can estimate the effective sampling size using two approaches [11, 12]:

- Reverse estimation uses samples from the flow

$$\mathrm{ESS} = \frac{1}{\mathbb{E}_q\left[w^2\right]} \approx \frac{1}{\frac{1}{N}\sum_{i=1}^{N}\hat{w}(x_i)^2}\,,\; x_i \sim q_\theta\,.$$

- Forward estimation uses samples from the target density p

$$\mathrm{ESS} = \frac{1}{\mathbb{E}_q\left[w^2\right]} = \frac{1}{\mathbb{E}_p\left[w\right]} \approx \frac{1}{\frac{1}{N}\sum_{i=1}^{N}\hat{w}(x_i)}\,,\; x_i \sim p\,.$$

As discussed in section 2, avoiding mode-collapse is of critical importance for NIS. However, the reverse estimator of the effective sampling size is not sensitive to mode-collapse as it just uses samples from the model. This is different for the forward estimator which in turn however has the disadvantage that it requires samples from the target density p which may be very costly to generate. It can therefore be challenging to detect mode-collapse—especially in situations for which MCMC methods fail.

For the particle in the double-well potential, we can use an overrelaxed Hybrid-Monte-Carlo algorithm to generate ground truth samples of the target p. These samples are then used to estimate the forward effective sampling size. This allows us to detect whether a certain training procedure leads to mode-collapse and thus quantify the degree of approximation correctly. The details of running the MCMC can be found in appendix E.

3.3. Discussion of results

3.3.1. Setup

we train a flow with RealNVP couplings for lattices of size $L \in \{8, 16, 32, 64\}$ for each value of the mass $m_0 \in \{2.75, 3.25\}$. We fix the other parameters to $\lambda = 1$ and $\mu^2 = -1$. For forward and reverse KL training, we use ReinfPQ (26) and the RepQP estimator (10) as the baseline respectively because these are the most widely used loss functions. We ensure that the baseline and path-gradient estimators use the same wall-time for training in order to ensure fair comparison and repeat training five times for uncertainty estimation. We then estimate the forward effective sampling size as described in the previous section. For a detailed description of the architecture and training procedure, we refer to the appendix E.

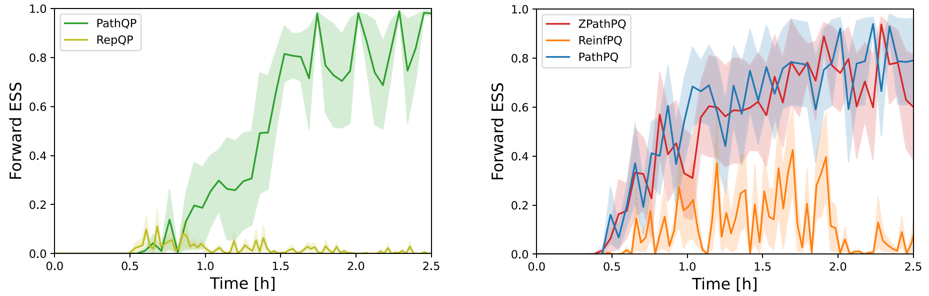

Figure 3. Left: training with the reverse KL estimators for $m_0 = 2.75$. Right: training with the forward KL estimators for $m_0 = 3.0$. The path gradient estimators clearly outperform their baselines. The bold line denotes the mean and the shaded area is the standard error over five training runs.

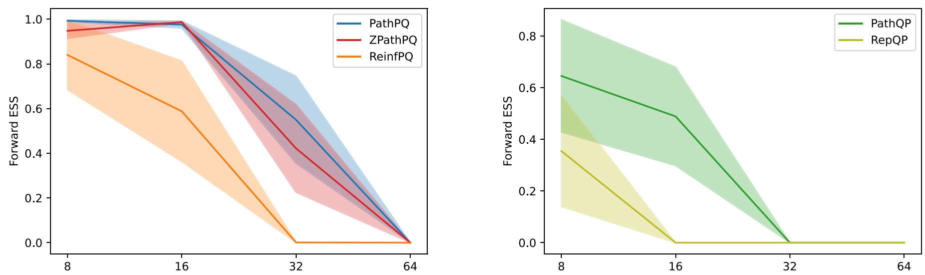

Figure 4. Forward ESS for $m_0 = 3.0$. Left: forward KL training using both path-gradient estimators (24) and (25) compared to the reinforce-based baseline (37). Right: reverse KL-training using the path-gradient estimator (15) compared to the base of the reparameterized estimator (10). The path-gradient-based estimators consistently outperform their baselines.

3.3.2. Approximation quality

Figure 3 shows an example of training as a function of wall-time. The path gradient estimators clearly outperform the standard ones for both forward and reverse KL divergence. Figure 4 compares the path-gradient estimators to a suitable baseline for other choices of the mass m_0 and demonstrates that the superior performance of the path gradient estimators is not due to a particular choice of the mass m_0. Our experiments confirm the observation of related work [22, 36], that optimizing the forward KL only works up to moderately high dimensions. Nevertheless using the path-wise gradients increases the feasible number of dimensions for which optimizing the forward KL is still a viable option.

3.3.3. Mode-collapse

in order to analyze mode-collapse, we compare the forward and reverse estimates of the effective sampling sizes for path-gradient training and its baseline. For table 1, we see that reverse training suffers from mode-collapse starting from relatively small lattice sizes, in contrast, to forward training as shown by the discrepancy between forward and reverse estimates of the effective sampling size. This underscores the superiority of forward training as it is crucial that mode-collapse is avoided for the statistical consistency of NIS. Furthermore, both for forward and reverse training, the path-gradient-based methods suffer substantially less from mode-collapse. In particular, we observe that the *Z*PathPQ estimator (25) seems to be less susceptible to mode-collapse. As we increase the mass m_0 and therefore move deeper into the broken phase, the mode-collapse becomes more severe across all methods, see table 2. Nevertheless, the path-gradient-based methods allow us to estimate at parameter values for which the standard methods fail.

3.3.4. Training phases

Figure 5 demonstrates that the *Z*PathPQ estimator (25) can suffer from vanishing gradients in the initial phase of training. This indeed confirms our theoretical analysis in section 2.2. It may therefore be advisable to start training with the PathPQ estimator (21). From figure 6, it can be seen that the norm of the PathQP estimator (15) indeed vanishes in the final phase of training while the standard RepQP estimator (10) is non-vanishing. This is to be expected since the latter contains the score term (13) which only vanishes in expectation.

Table 1. Results of training a real-valued non-volume preserving (RealNVP) normalizing flow for $m_0 = 3$. Note that only the forward estimators can detect mode-collapse as discussed in section 3.2. As a result, a large value of the reverse with a corresponding small value for the forward effective sample size (ESS) is a clear indication of mode-collapse. The path-gradient estimators therefore not only consistently outperform the baselines but also are significantly more robust to mode-collapse.

ESS	*d*	ReinfPQ	ZPathPQ	PathPQ	RepQP	PathQP
FW ESS	8	**0.84** ± .14	**0.95** ± .03	**0.99** ± .00	0.03 ± .02	**0.65** ± .19
	16	**0.59** ± .20	**0.99** ± .00	**0.98** ± .01	0.00 ± .00	0.49 ± .17
	32	0.00 ± .00	**0.42** ± .18	**0.55** ± .17	0.00 ± .00	0.00 ± .00
	64	**0.00** ± .00	**0.00** ± .00	**0.00** ± .00	**0.00** ± .00	**0.00** ± .00
Rev ESS	8	1.00 ± .00	**1.00** ± .00	**1.00** ± .00	0.99 ± .00	**1.00** ± .00
	16	0.99 ± .00	**0.99** ± .00	**1.00** ± .00	0.06 ± .04	0.99 ± .00
	32	**0.96** ± .00	**0.98** ± .00	**0.98** ± .00	0.78 ± .13	0.80 ± .07
	64	0.93 ± .01	0.95 ± .00	0.92 ± .03	0.90 ± .04	**0.97** ± .00

Note: Best results within a statistical significance of p-value < 0.05, according to the Wilcoxon tests, are shown in bold.

Table 2. Same as table 1 but for a relatively large mass of $m_0 = 3.25$. At this point in parameter space, the modes of the distribution are separated by a pronounced action barrier which can lead to mode collapse.

ESS	*d*	ReinfPQ	ZPathPQ	PathPQ	RepQP	PathQP
FW ESS	8	0.40 ± .22	**0.81** ± .16	**1.00** ± .00	0.00 ± .00	0.18 ± .16
	16	0.00 ± .00	**0.86** ± .10	**0.79** ± .18	0.00 ± .00	0.00 ± .00
	32	0.00 ± .00	**0.14** ± .13	0.00 ± .00	0.00 ± .00	0.00 ± .00
	64	**0.00** ± .00	**0.00** ± .00	**0.00** ± .00	**0.00** ± .00	**0.00** ± .00
Rev ESS	8	0.99 ± .00	**1.00** ± .00	**1.00** ± .00	**0.77** ± .17	0.93 ± .06
	16	0.99 ± .00	**0.99** ± .00	**0.99** ± .00	**0.99** ± .00	**0.60** ± .16
	32	**0.92** ± .03	**0.96** ± .01	**0.97** ± .01	**0.98** ± .00	**0.91** ± .06
	64	0.94 ± .00	0.95 ± .00	0.96 ± .00	0.87 ± .06	**0.98** ± .00

Note: Best results within a statistical significance of p-value < 0.05, according to the Wilcoxon tests, are shown in bold.

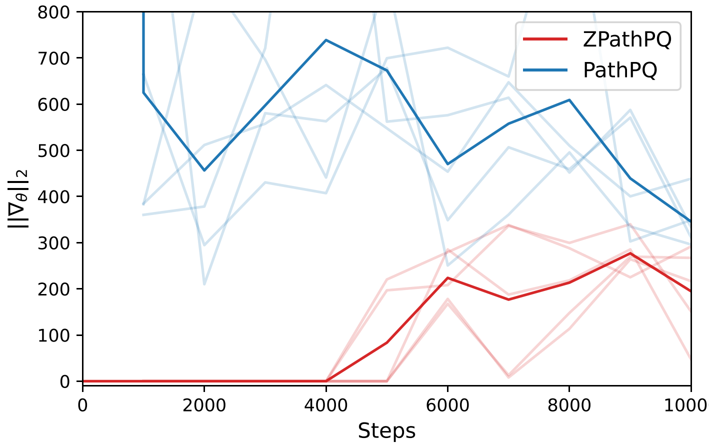

Figure 5. Norm of the *Z*PathPQ and PathPQ estimators in the initial phase of training averaged over 5 training runs for the Double-Well with 64 timesteps and $m_0 = 2.75$. This demonstrates that the *Z*PathPQ estimator can lead to vanishing gradients in this initial training phase—as expected from equation (30).

3.3.5. Runtime

As discussed in section 2.3, our estimator for the path gradient has the same memory requirements as the standard estimator for the total derivative. This is crucial as it allows us to train with the same batch-size. However, we expect double runtime per iteration. This is indeed confirmed by our numerical experiments, see figure 7. In order to account for this, the training using the PathQP and ReinfPQ baselines are therefore run for twice the number of iterations. This ensures a fair comparison as all estimators have the same overall training time.

3.4. Estimation of thermodynamic observables

A crucial advantage of sampling with normalizing flows is that they allow us to estimate thermodynamic observables, such as the free energy[7] $F = -\ln Z$, at specific points in parameter space [7, 12]. This is in stark

[7] We note that we have adopted a normalization of the free energy such that its temperature dependence is one in order to alleviate notation.

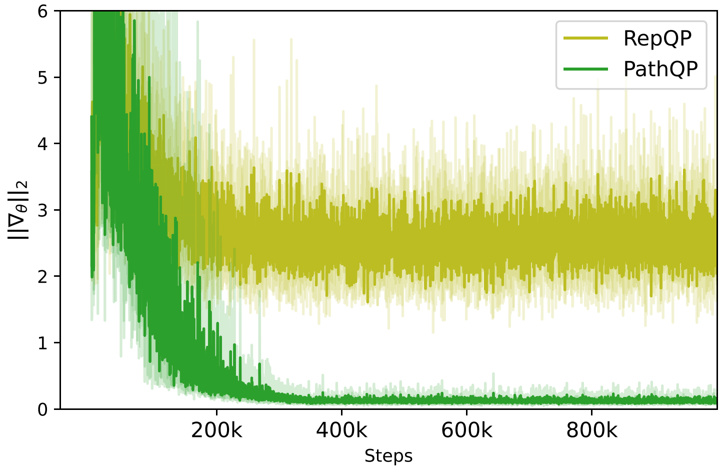

Figure 6. The norm of the gradient estimated by PathQP goes to zero when the target density p is well approximated towards the end of the training. The norms of the gradients are averaged over three runs for the Double-Well experiments with eight timesteps and $m_0 = 2.75$ in equation (33).

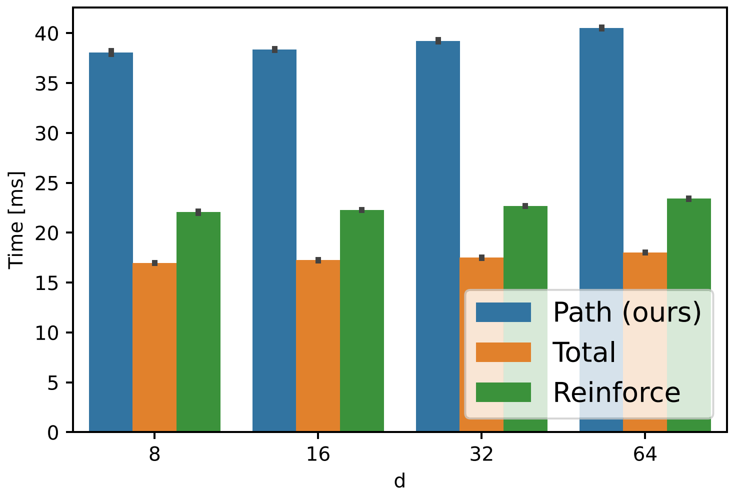

Figure 7. Runtime per iteration for calculation of the path, total, and reinforce gradient. The path-gradient has roughly twice the computational cost at the same memory requirements. This overhead is more than compensated by faster overall training convergence.

contrast to established MCMC methods which only allow to estimate free energy differences. For this, it is however crucial that the normalizing flow is trained in a manner that avoids mode-collapse. Free energy estimation, therefore, represents an ideal setting to illustrate the downstream advantages of path-gradient training.

As shown in Nicoli *et al* [7, 12], a point estimate of the free energy can be obtained by

$$\hat{F}_q = -\ln\left(\frac{1}{N}\sum_{i=1}^{N}\tilde{w}(x_i)\right), \quad x_i \sim q_\theta\,. \tag{34}$$

which is sensitive to mode-collapse since estimation is based on samples from the flow. As a ground-truth proxy, we also estimate the free energy by

$$\hat{F}_p = \ln\left(\frac{1}{N}\sum_{i=1}^{N}\frac{q_\theta(x_i)}{\exp(-S(x_i))}\right), \quad x_i \sim p\,. \tag{35}$$

which requires an expensive MCMC simulation to generate samples from the target p. On the other hand, this is insensitive to mode-collapse. We refer to [7, 12] for more details.

Figure 8 shows the difference $F_q - F_p$ between the free energy estimators at various points in parameter space. We observe that path gradient training significantly improves the estimation accuracy due to its ability

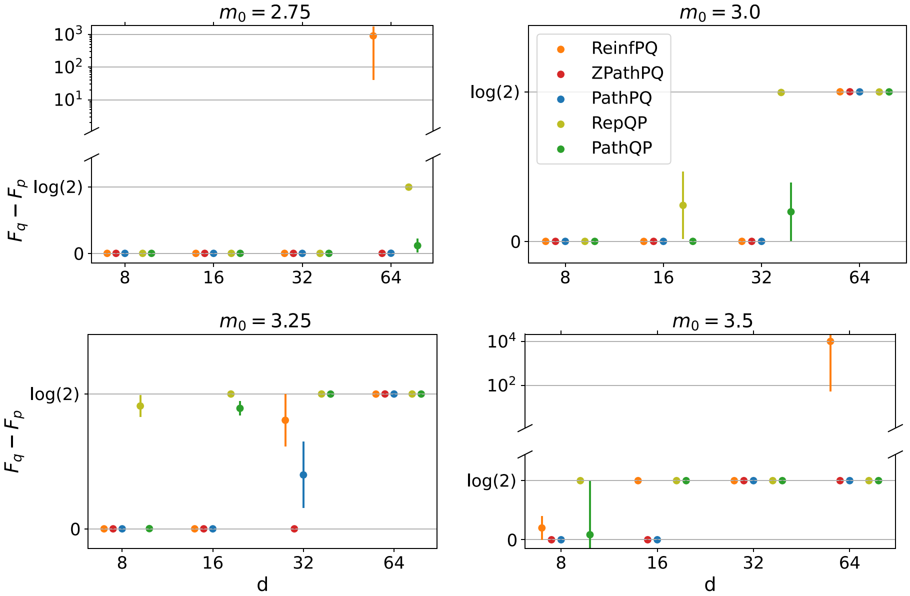

Figure 8. Difference between the two estimators $\hat{F}_p$ and $\hat{F}_q$ for the free energy. The estimator $\hat{F}_q$ uses samples from the variational distribution q_θ while $\hat{F}_p$ uses samples from the overrelaxed HMC. The estimators both use 1 million samples. If both estimators give consistent results, i.e. $\hat{F}_p \approx \hat{F}_q$, there is no sign of mode dropping. If $\hat{F}_p$ is larger than $\hat{F}_q$ by a value of $\log(2)$, this indicates that q only captures half of the modes of p. A larger value than $\log(2)$ indicates that the variational distribution q failed to cover the relevant regions of the target density p.

to alleviate mode-collapse. As expected, this trend is consistent with the experimental results for the effective sampling size in the previous section.

4. Conclusion

Path-gradient estimators bring substantial improvements for training normalizing flows in the context of NIS. In this work, we have proposed an algorithm to calculate path gradient estimators that can be applied as a drop-in replacement of standard estimators for any invertible normalizing flow. Crucially, our algorithm has the same memory requirements as standard approaches and thus allows us to use the same batch size for training, which is essential for training with MC methods. Furthermore, the lower variance of the path-wise gradient estimators not only leads to faster convergence during training but also better overall approximation quality.

The path-wise gradient estimators allow us to apply the flows to higher-dimensional problems by pushing the limit up to which the inclusive forward KL is applicable. Our experiments have demonstrated the favorable behavior of the forward KL with respect to mode collapse in moderately high dimensions, which enables us to tackle mode collapse without any prior domain knowledge of the problem at hand. We have analyzed our estimators theoretically and shown that they have lower variance in the limit of perfect approximation. We furthermore theoretically compared the properties of the forward estimators in the initial and final phases of training. We expect that the estimator proposed in this work will become the new de-facto standard for training normalizing flows on variational inference tasks due to its superior performance and implementation simplicity.

For future work, it would be interesting to theoretically prove the lower variance of the path-gradient estimators off the limit of perfect approximation as is strongly suggested by our experiments. Furthermore, it would be very desirable to construct an estimator which is as fast as the one derived in [14] since their proposal is more performance than ours but unfortunately completely limited to the special case of a CNF.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

K A N, S N and P K are supported by the German Ministry for Education and Research (BMBF) as BIFOLD—Berlin Institute for the Foundations of Learning and Data under Grant Nos. 01IS18025A and 01IS18037A. K A N acknowledges the organizers of the Summer School in Machine Learning for Quantum Physics and Chemistry (Warsaw 2021). The publication fees of this manuscript were covered by the 'best poster prize' awarded to the author during the school.

Appendix A. Derivation of the estimator for forward KL

A natural approach to estimating the gradient of the forward KL

$$\frac{d}{d\theta}\mathrm{KL}(p,q_\theta)=\frac{d}{d\theta}\mathbb{E}_{x\sim p}\left[\log\left(\frac{p(x)}{q_\theta(x)}\right)\right], \tag{36}$$

is to re-weight the expectation value with respect to the target p such that it becomes an expectation with respect to q

$$\begin{aligned}\frac{d}{d\theta}\mathrm{KL}(p,q_\theta)&=\mathbb{E}_{x\sim p}\left[\frac{\partial}{\partial\theta}\log\left(\frac{p(x)}{q_\theta(x)}\right)\right]\\&=\mathbb{E}_{x\sim q_\theta}\left[\frac{p(x)}{q_\theta(x)}\frac{\partial}{\partial\theta}\log\left(\frac{p(x)}{q_\theta(x)}\right)\right] &(37)\\&=\mathbb{E}_{x\sim q_\theta}\left[\frac{\partial}{\partial\theta}\frac{p(x)}{q_\theta(x)}\right]. &(38)\end{aligned}$$

We then use the reparameterization trick to obtain

$$\frac{d}{d\theta}\mathrm{KL}(p,q_\theta)=\mathbb{E}_{z\sim q_Z}\left[\left.\frac{\partial}{\partial\theta}\frac{p(x)}{q_\theta(x)}\right|_{x=g_\theta(z)}\right].$$

Using the relation (11) of the partial derivative in (38) with the path and total derivative, this can be rewritten as

$$\frac{d}{d\theta}\mathrm{KL}(p,q_\theta)=\mathbb{E}_{z\sim q_Z}\left[\frac{d}{d\theta}\frac{p(g_\theta(z))}{q_\theta(g_\theta(z))}-\blacktriangledown_\theta\frac{p(g_\theta(z))}{q_\theta(g_\theta(z))}\right].$$

The first summand on the right-hand-side is vanishing because

$$\mathbb{E}_{z\sim q_Z}\left[\frac{d}{d\theta}\frac{p(g_\theta(z))}{q_\theta(g_\theta(z))}\right]=\frac{d}{d\theta}\mathbb{E}_{z\sim q_Z}\left[\frac{p(g_\theta(z))}{q_\theta(g_\theta(z))}\right]=\frac{d}{d\theta}\mathbb{E}_{x\sim q_\theta}\left[\frac{p(x)}{q_\theta(x)}\right]=0,$$

where we have used in the last step that $\mathbb{E}_{x\sim q_\theta}\left[\frac{p(x)}{q_\theta(x)}\right]=\int \mathrm{d}^d x\, p(x)=1$.

We have thus obtained the expression of the forward KL gradient in terms of a path-derivative given in the main text, i.e.

$$\frac{d}{d\theta}\mathrm{KL}(p,q_\theta)=-\mathbb{E}_{z\sim q_Z}\left[\blacktriangledown_\theta\frac{p(g_\theta(z))}{q_\theta(g_\theta(z))}\right] \tag{39}$$

We note that this relation can also be derived using the so-called DReG-identity of Tucker *et al* [15].

Appendix B. Variance of the reinforce estimator

The ReinfPQ estimator was defined in (26) as

$$\begin{aligned}\frac{d}{d\theta}\mathrm{KL}(p,q_\theta)&\approx\frac{1}{N}\sum_{i=1}^{N}\frac{\tilde{w}_i}{\hat{Z}}\frac{\partial}{\partial\theta}\log(\tilde{w}_i) &(40)\\&=\frac{1}{N}\sum_{i=1}^{N}\frac{e^{-S(x_i)}}{q_\theta(x_i)\hat{Z}}\frac{\partial}{\partial\theta}\log q_\theta(x_i).\end{aligned}$$

Its second moment is therefore given by

$$\frac{1}{N}\mathbb{E}_{x\sim q_\theta}\left[\hat{w}(x)^2\left(\frac{\partial}{\partial\theta}\log q_\theta(x)\frac{\partial}{\partial\theta}\log q_\theta(x)\right)\right].$$

In the limit of perfect approximation, i.e. $q_\theta(x)=p(x)$ for any $x\in\mathcal{X}$, it holds that $\hat{w}(x)=1$ and thus the covariance of the ReinfPQ estimator converges to the Fisher information of the variational distribution

$$\frac{1}{N}\underbrace{\mathbb{E}_{x\sim q_\theta}\left[\frac{\partial}{\partial\theta}\log q_\theta(x)\frac{\partial}{\partial\theta}\log q_\theta(x)\right]}_{=\mathcal{I}(\theta)}=\frac{1}{N}\mathcal{I}(\theta)\,, \tag{41}$$

and thus is generically non-vanishing even in the limit of perfect approximation.

Appendix C. Asymptotic behavior of path gradient estimators of forward KL divergence

Here, we analyze the bias and variance of the PathPQ (24) and ZPathPQ (25) estimators, which are defined as

$$\mathrm{PathPQ}_N=-\frac{1}{N}\sum_{i=1}^{N}\frac{\tilde{w}_i}{\frac{1}{N}\sum_{j=1}^{N}\tilde{w}_j}\boldsymbol{\nabla}_\theta\log\tilde{w}_i,$$
$$\mathrm{ZPathPQ}_N=\mathrm{PathPQ}_N+\frac{1}{N^2}\sum_{i=1}^{N}\left(\frac{\tilde{w}_i}{\frac{1}{N}\sum_{j=1}^{N}\tilde{w}_i}\right)^2\boldsymbol{\nabla}_\theta\log\tilde{w}_i.$$

Throughout this appendix, we use a shorthand notation for $\tilde{w}_i=\tilde{w}(x_i)=\tilde{w}(g_\theta(z_i))$. Using $w_i=\tilde{w}_i/Z=p(x_i)/q_\theta(x_i)$, we can rewrite the estimators above as

$$\mathrm{PathPQ}_N=-\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\frac{1}{N}\sum_{j=1}^{N}w_j}\boldsymbol{\nabla}_\theta w_i, \tag{42}$$

$$\mathrm{ZPathPQ}_N=\mathrm{PathPQ}_N+\frac{1}{N^2}\sum_{i=1}^{N}\frac{w_i}{\left(\frac{1}{N}\sum_{j=1}^{N}w_i\right)^2}\boldsymbol{\nabla}_\theta w_i, \tag{43}$$

where we used $w_i\boldsymbol{\nabla}_\theta\log w_i=\boldsymbol{\nabla}_\theta w_i$. Note that $\mathbb{E}_{q_\theta}[w]=1$.

C.1. Bias

Let $\epsilon\equiv\frac{1}{N}\sum_{i=1}^{N}(1-w_i)$. Then, $\mathbb{E}_{q_\theta}[\epsilon]=0$ and $\epsilon=\mathcal{O}_p(N^{-1/2})$, and therefore

$$\frac{1}{\frac{1}{N}\sum_{j=1}^{N}w_j}=(1-\epsilon)^{-1}=1+\epsilon+\epsilon^2+\epsilon^3+\mathcal{O}_p(N^{-2}). \tag{44}$$

Since $\{w_i\}_{i=1}^{N}$ are independent, the following hold for any function $\kappa(\cdot)$:

$$\begin{aligned}\mathbb{E}_{q_\theta}[\kappa(w_i)\epsilon]&=\mathbb{E}_{q_\theta}\left[\kappa(w_i)\frac{1}{N}\sum_{j=1}^{N}(1-w_j)\right]\\&=\frac{1}{N}\mathbb{E}_{q_\theta}[\kappa(w_i)(1-w_i)],\end{aligned} \tag{45}$$

$$\begin{aligned}\mathbb{E}_{q_\theta}[\kappa(w_i)\epsilon^2]&=\mathbb{E}_{q_\theta}\left[\kappa(w_i)\frac{1}{N^2}\left(\sum_{j=1}^{N}\sum_{k=1}^{N}(1-w_j)(1-w_k)\right)\right]\\&=\mathbb{E}_{q_\theta}\left[\kappa(w_i)\frac{1}{N^2}\left((1-w_i)^2+\sum_{j\neq i}(1-w_j)^2+\sum_{j=1}^{N}\sum_{k\neq j}(1-w_j)(1-w_k)\right)\right]\\&=\mathbb{E}_{q_\theta}\left[\kappa(w_i)\frac{1}{N^2}\left((1-w_i)^2+(N-1)\mathbb{E}_{w\sim q_\theta}[(1-w)^2]\right)\right]\\&=\frac{1}{N}\mathbb{E}_{w\sim q_\theta}\left[\kappa(w_i)\right]\mathbb{E}_{w\sim q_\theta}[(1-w)^2]+\mathbb{E}_{q_\theta}\left[\kappa(w_i)\cdot\mathcal{O}_p(N^{-2})\right],\end{aligned} \tag{46}$$

$$\begin{aligned}\mathbb{E}_{q_\theta}[\kappa(w_i)\epsilon^3] &= \mathbb{E}_{q_\theta}\left[\kappa(w_i)\frac{1}{N^p\,3}\left((1-w_i)^3+\sum_{j\neq i}((1-w_j)^3+3(1-w_i)(1-w_j)^2)\right)\right]\\ &= \mathbb{E}_{q_\theta}\left[\kappa(w_i)\frac{1}{N^3}\left((1-w_i)^3+(N-1)\mathbb{E}_{w\sim q_\theta}\left[((1-w)^3+3(1-w_i)(1-w)^2)\right]\right)\right]\\ &= \mathbb{E}_{q_\theta}\left[\kappa(w_i)\cdot\mathcal{O}_p(N^{-2})\right]. \end{aligned} \tag{47}$$

C.1.1. PathPQ

By using equations (44)–(47), the bias of the PathPQ estimator (42) from the true gradient $-\mathbb{E}_{w\sim q_\theta}[\blacktriangledown_\theta w]$ is evaluated as

$$\begin{aligned}&\mathbb{E}_{q_\theta}[\mathrm{PathPQ}_N]+\mathbb{E}_{w\sim q_\theta}[\blacktriangledown_\theta w]\\ &\quad= -\frac{1}{N}\left(\mathbb{E}_{w\sim q_\theta}[(1-w)\blacktriangledown_\theta w]+\mathbb{E}_{w\sim q_\theta}\left[(1-w)^2\right]\mathbb{E}_{q_\theta}[\blacktriangledown_\theta w]\right)+\mathbb{E}_{w\sim q_\theta}\left[\mathcal{O}_p(N^{-2})\cdot\blacktriangledown_\theta w\right].\end{aligned} \tag{48}$$

The bias of the PathPQ estimator is of N^{-1} times smaller scale than the true gradient. However, the leading term vanishes in the converging phase where $q_\theta(x)\approx p(x)$ holds. For example, if $w=1+\mathcal{O}_p(N^{-1})$, the bias is of N^{-2} times smaller scale. We suppose that this, however, does not affect the practical training performance, because the estimation error is dominated by the variance, as will be shown in appendix C.2.

C.1.2. ZPathPQ

Similarly, by using equations (44)–(47), the expectation value of the second term of the ZPathPQ estimator (43) is evaluated as

$$\mathbb{E}_{q_\theta}\left[\frac{1}{N^2}\sum_{i=1}^{N}\frac{w_i}{\left(\frac{1}{N}\sum_{j=1}^{N}w_i\right)^2}\blacktriangledown_\theta w_i\right]=\frac{1}{N}\mathbb{E}_{w\sim q_\theta}[w\blacktriangledown_\theta w]+\mathbb{E}_{w\sim q_\theta}\left[\mathcal{O}_p(N^{-2})\cdot\blacktriangledown_\theta w\right].$$

Therefore the bias of the ZPathPQ estimator is given as

$$\begin{aligned}&\mathbb{E}_{q_\theta}[\mathrm{ZPathPQ}_N]+\mathbb{E}_{w\sim q_\theta}[\blacktriangledown_\theta w]\\ &\quad= -\frac{1}{N}\left(\mathbb{E}_{w\sim q_\theta}[(1-2w)\blacktriangledown_\theta w]+\mathbb{E}_{w\sim q_\theta}\left[(1-w)^2\right]\mathbb{E}_{q_\theta}[\blacktriangledown_\theta w]\right)+\mathbb{E}_{w\sim q_\theta}\left[\mathcal{O}_p(N^{-2})\cdot\blacktriangledown_\theta w\right].\end{aligned} \tag{49}$$

A notable difference from the PathPQ estimator is that the leading order term does not vanish when $w\approx 1$, and therefore the bias is always N^{-1} times smaller than the true gradient.

C.2. Variance

We analyze the asymptotic behavior of the variance of gradient estimators by using the Delta Method (see e.g. theorem 5.5.28 in Casella and Berger [44] or paragraph 4.9 in Small [45]):

Theorem 1 (**[45]**). *Assume that $f(x,y,z)$ is a differentiable function, and $\{(X_i,Y_i,Z_i)\}_{i=1}^N$ are independently and identically distributed. Assume that the distribution of (X_i,Y_i,Z_i) satisfies the conditions for the central limit theorem, and hence $\overline{X}=\frac{1}{N}\sum_{i=1}^N X_i$, $\overline{Y}=\frac{1}{N}\sum_{i=1}^N Y_i$, and $\overline{Z}=\frac{1}{N}\sum_{i=1}^N Z_i$ are normally distributed in the asymptotic limit. Then it holds that*

$$\begin{aligned}Var(f(\overline{X},\overline{Y},\overline{Z})) &= f_x^2 Var(\overline{X})+f_y^2 Var(\overline{Y})+f_z^2 Var(\overline{Z})\\ &\quad+2f_xf_yCov(\overline{X},\overline{Y})+2f_yf_zCov(\overline{Y},\overline{Z})+2f_zf_xCov(\overline{Z},\overline{X})\\ &\quad+\mathcal{O}(N^{-2}),\end{aligned} \tag{50}$$

where

$$(f_x,f_y,f_z)=\left(\frac{\partial f}{\partial x},\frac{\partial f}{\partial y},\frac{\partial f}{\partial z}\right)\Big|_{(x,y,z)=(\mathbb{E}[\overline{X}],\mathbb{E}[\overline{Y}],\mathbb{E}[\overline{Z}])}$$

are the derivatives evaluated at $(x,y,z)=(\mathbb{E}[\overline{X}],\mathbb{E}[\overline{Y}],\mathbb{E}[\overline{Z}])$.

C.2.1. PathPQ

The PathPQ estimator (42) is a ratio estimator with two variables as

$$\mathrm{PathPQ}_N \equiv f(\overline{X},\overline{Y}) = -\underbrace{\frac{1}{\frac{1}{N}\sum_{j=1}^{N} w_j}}_{1/\overline{Y}}\underbrace{\frac{1}{N}\sum_{i=1}^{N} \blacktriangledown_\theta w_i}_{\overline{X}} = -\frac{\overline{X}}{\overline{Y}}.$$

Since

$$\begin{aligned}
f_x &= -\frac{1}{\mathbb{E}[\overline{Y}]} = -\frac{1}{\mathbb{E}_{q_\theta}[w]} = -1,\\
f_y &= \frac{\mathbb{E}[\overline{X}]}{\mathbb{E}[\overline{Y}^2]} = \frac{\mathbb{E}_{q_\theta}[\blacktriangledown_\theta w]}{\mathbb{E}_{q_\theta}[w]^2},\\
\mathrm{Var}[\overline{X}] &= \frac{1}{N}\mathrm{Var}_{q_\theta}[\blacktriangledown_\theta w],\\
\mathrm{Var}[\overline{Y}] &= \frac{1}{N}\mathrm{Var}_{q_\theta}[w],\\
\mathrm{Cov}[\overline{X},\overline{Y}] &= \frac{1}{N}\mathrm{Cov}_{q_\theta}[\blacktriangledown_\theta w, w],
\end{aligned}$$

Equation (50) gives

$$\begin{aligned}
\mathrm{Var}(\mathrm{PathPQ}_N) &= \mathrm{Var}(f(\overline{X},\overline{Y}))\\
&= \frac{1}{\mathbb{E}[\overline{Y}]^2}\mathrm{Var}(\overline{X}) + \frac{\mathbb{E}[\overline{X}]^2}{\mathbb{E}[\overline{Y}]^4}\mathrm{Var}[\overline{Y}] - 2\frac{1}{\mathbb{E}[\overline{Y}]}\frac{\mathbb{E}[\overline{X}]}{\mathbb{E}[\overline{Y}]^2}\mathrm{Cov}[\overline{X},\overline{Y}] + O(N^{-2})\\
&= \frac{1}{N}\mathrm{Var}_{q_\theta}[\blacktriangledown_\theta w] + \frac{\mathbb{E}_{q_\theta}[\blacktriangledown_\theta w]^2}{N}\mathrm{Var}_{q_\theta}[w]\\
&\quad - 2\frac{\mathbb{E}_{q_\theta}[\blacktriangledown_\theta w]}{N}\mathrm{Cov}_{q_\theta}[\blacktriangledown_\theta w, w] + \mathcal{O}(N^{-2}).
\end{aligned} \tag{51}$$

C.2.2. ZPathPQ

ZPathPQ (42) has an additional term to PathPQ, i.e.

$$\mathrm{ZPathPQ}_N \equiv f'(\overline{X},\overline{Y},\overline{Z}) = f(\overline{X},\overline{Y}) + g(\overline{Y},\overline{Z}), \qquad \text{where}$$

$$g(\overline{Y},\overline{Z}) = \underbrace{\frac{1}{\left(\frac{1}{N}\sum_{j=1}^{k} w_j\right)^2}}_{\overline{Y}^{-2}}\underbrace{\frac{1}{N^2}\sum_{i=1}^{k} w_i wi \blacktriangledown_\theta w_i}_{\overline{Z}} = \frac{\overline{Z}}{\overline{Y}^2}.$$

Since

$$\begin{aligned}
f'_x &= f_x,\\
f'_y &= f_y + 2\frac{\mathbb{E}[\overline{Z}]}{\mathbb{E}[\overline{Y}^3]} = \frac{\mathbb{E}_{q_\theta}[\blacktriangledown_\theta w]}{\mathbb{E}_{q_\theta}[w]^2} + \frac{2}{N}\frac{\mathbb{E}_{q_\theta}[w\blacktriangledown_\theta w]}{E_{q_\theta}[w]^3},\\
f'_z &= -\frac{1}{\mathbb{E}[\overline{Y}]^2} = -\frac{1}{\mathbb{E}_{q_\theta}[w]^2},\\
\mathrm{Var}(\overline{Z}) &= \frac{1}{N^3}\mathrm{Var}_{q_\theta}(w\blacktriangledown_\theta w),\\
\mathrm{Cov}(\overline{Y},\overline{Z}) &= \frac{1}{N^2}\mathrm{Cov}_{q_\theta}[w, w\blacktriangledown_\theta w],\\
\mathrm{Cov}(\overline{Z},\overline{X}) &= \frac{1}{N^2}\mathrm{Cov}_{q_\theta}(w\blacktriangledown_\theta w,\ \blacktriangledown_\theta w),
\end{aligned}$$

Equation (50) gives

$$\begin{aligned}\mathrm{Var}(\mathrm{ZPathPQ}_N) &= \mathrm{Var}(f'(\overline{X},\overline{Y},\overline{Z}))\\ &= \mathrm{Var}(\overline{X})f'^2_X + \mathrm{Var}(\overline{Y})f'^2_Y + \mathrm{Var}(\overline{Z})f'^2_z\\ &\quad + 2f'_x f'_y \mathrm{Cov}(\overline{X},\overline{Y}) + 2f'_y f'_z \mathrm{Cov}(\overline{Y},\overline{Z}) + 2f'_z f'_x \mathrm{Cov}(\overline{Z},\overline{X})\\ &\quad + \mathcal{O}(N^{-2})\\ &= \frac{1}{N}\mathrm{Var}_{q_\theta}[\blacktriangledown_\theta w] + \frac{\mathbb{E}_{q_\theta}[\blacktriangledown_\theta w]^2}{N}\mathrm{Var}_{q_\theta}[w]\\ &\quad - 2\frac{\mathbb{E}_{q_\theta}[\blacktriangledown_\theta w]}{N}\mathrm{Cov}_{q_\theta}[\blacktriangledown_\theta w, w] + \mathcal{O}(N^{-2}).\end{aligned} \tag{52}$$

We see that PathPQ and ZPathPQ have the same leading order variance (compare (51) and (52)). This is because the difference $g(\overline{Y},\overline{Z})$ between the two estimators is $O_p(N^{-1})$ and thus its contribution to the variance is $O(N^{-2})$.

C.3. Summary

Our analysis revealed that

- Both of the PathPQ and the ZPathPQ estimators have the biases, (48) and (49), that are of N^{-1} times smaller order than the true gradient. However, when the model distribution q_θ approaches the target distribution p in the converging phase of training, the leading order bias vanishes for PathPQ, while it stays for ZPathPQ. Therefore, we can say that PathPQ has a smaller bias than ZPathPQ in the converging phase.
- The PathPQ and ZPathPQ have the same leading order variances, (51) and (52), which is N^{-1} times smaller (hence the standard deviation is $N^{-1/2}$ times smaller) order than the sample gradients.
- For both estimators, the estimation error is dominated by the standard deviation, and therefore, we suppose that the advantage of PathPQ in terms of the bias in the converging phase does not have a large effect on the training performance.

We conclude that both estimators should perform similarly in the converging phase.

Appendix D. Initial training phase behavior of path gradient estimators of forward KL divergence

Weight degeneracy [38] can become a serious issue for importance sampling. Weight degeneracy is the phenomenon of only a few importance weights taking high values, while the other degenerate weights take negligible values. At the start of training in high-dimensional problem settings, it is often the case in practice that only a singular sample is non-degenerate. Here we will show this phenomenon is problematic for the ZPathPQ estimator.

In order to analyze this initial training regime theoretically, we assume without loss of generality that the N-th sample is singular, i.e.

$$\frac{p(x_i)}{p(x_N)} = \mathcal{O}(\epsilon), \quad \frac{\|\nabla_{x_i} p(x_i)\|}{p(x_N)} = \mathcal{O}(\epsilon) \qquad \text{for } i = 1,\ldots,N-1, \tag{53}$$

and

$$q_\theta(x_i) = \mathcal{O}(1), \quad \|\nabla_x q_\theta(x_i)\| = \mathcal{O}(1) \qquad \text{for } i = 1,\ldots,N, \tag{54}$$

for small $\varepsilon > 0$. The former assumption (53) comes from the fact that most interesting densities in physics have an exponential fall-off around their modes, while the latter assumption (54) comes from the fact that all samples $\{x_i\}_{i=1}^N$ are generated from the sampler q_θ. We also assume that $\epsilon \ll N^{-1}, d$, and ignore the scaling w.r.t. N and d.

Let $\tau = p(x_N)$[8]. Then, the assumptions (53) and (54) lead to

$$\begin{aligned}\nabla_{x_i} \frac{p(x_i)}{q(x_i)} &= \nabla_{x_i}(p(x_i))\frac{1}{q(x_i)} - \frac{p(x_i)}{q(x_i)^2}\nabla_{x_i} q(x_i) \\ &= \mathcal{O}(\tau\epsilon)\frac{1}{q(x_i)} - \mathcal{O}(\tau\epsilon)\nabla_{x_i} q(x_i) \\ &= \mathcal{O}(\tau\epsilon)\end{aligned}$$

for $i = 1, \ldots, N-1$. Using the mild assumption that $\frac{\partial x}{\partial \theta}$ does not diverge, the corresponding path-wise gradient is in the same order:

$$\blacktriangledown_\theta w(x_i) = \frac{\partial w(x_i)}{\partial x_i}\frac{\partial x_i}{\partial \theta} = \mathcal{O}(\tau\epsilon). \tag{55}$$

D.1. PathPQ

Under the singularity assumptions, (53) and (54), the PathPQ estimator (42) can be evaluated as

$$\begin{aligned}\mathrm{PathPQ}_N &= \sum_{i=1}^{N} \frac{1}{\sum_{j=1}^{N} w_j} \blacktriangledown_\theta w_i \\ &= \frac{1}{\sum_{j=1}^{N} w_j}\left(\sum_{i=1}^{N-1} \blacktriangledown_\theta w_i + \blacktriangledown_\theta w_N\right) \\ &= \frac{1}{w_N}(1 - \mathcal{O}(\epsilon))(\blacktriangledown_\theta w_N + \mathcal{O}(\epsilon)) \\ &= \frac{\blacktriangledown_\theta w_N}{w_N} + \mathcal{O}(\epsilon).\end{aligned}$$

Therefore, if $\|\nabla_{x_N} p(x_N)\| = O(\tau)$ and hence $\frac{\blacktriangledown_\theta w_N}{w_N} = O(1)$, the gradient direction is dominated by the singular (N-th) sample. Interestingly, the dominating term for the PathPQ gradient is proportional to the dominating term for the PathQP gradient:

$$\mathrm{PathQP}_N = \frac{1}{N}\sum_{i=1}^{N} \blacktriangledown_\theta w_i = \frac{1}{N}\blacktriangledown w_N + \mathcal{O}(\tau\epsilon).$$

This implies that the PathPQ and the PathQP gradient behave similarly in the initial training phase, and could point to an explanation for the observation of Geffner and Domke [23] that their path-wise gradient estimators for alpha-divergences—such as the forward KL—seem to optimize the reverse KL in a high dimensional setting.

D.2. ZPathPQ

Under the same singularity assumptions, (53) and (54), the ZPathPQ estimator (43) on the other hand does not have an order one contribution:

$$\begin{aligned}\mathrm{ZPathPQ}_N &= \sum_{i=1}^{N}\left(\frac{1}{\sum_{j=1}^{N} w_j} - \frac{w_i}{\left(\sum_{j=1}^{N} w_j\right)^2}\right)\blacktriangledown_\theta w_i \\ &= \left(\frac{1}{\sum_{j=1}^{N} w_j} - \frac{w_N}{\left(\sum_{j=1}^{N} w_j\right)^2}\right)\blacktriangledown_\theta w_N + \mathcal{O}(\epsilon) \\ &= \left(\frac{1}{w_N} - \frac{w_N}{w_N^2}\right)\blacktriangledown_\theta w_N + \mathcal{O}(\epsilon) \\ &= \mathcal{O}(\epsilon).\end{aligned}$$

The ZPathPQ estimator is thus expected to struggle in the initial training phase with its weak gradient signal.

[8] Note that it is assumed that $\tau = O(1)$ in section 2.2 for simplicity.

Figure 5 empirically validates this behavior. This might also explain why the ZPathPQ estimator shows unstable behavior in optimizing VAEs (see e.g. experiments on structured MNIST in Tucker *et al* [15])[9].

Appendix E. Experimental details

E.1. Double-well

For the quantum mechanical particle in the double-well potential, we trained a flow with RealNVP couplings which uses tanh activation, eight coupling layers with three fully connected layers with a width of 200 neurons each. The batch size was 4000. We used adaptive Moment estimation (ADAM) with an initial learning rate of 5×10^{-5}, $\beta=(0.9,0.999)$. The learning rate was decreased using a plateau learning rate schedule with a patience of 3000 to a minimum of 10^{-7}. We used gradient norm clipping with a $l2$ norm and a max norm of 1.0. The base distribution for the normalizing flow was a univariate normal distribution with a standard deviation of 10. The baselines estimators were run for 200k and 170k epochs, while the path gradient estimators were run for 100k epochs, this ensured that the wall-time duration for training the path-wise gradient estimators did not exceed the duration of training the baselines. The training was done on NVidia P100 GPUs.

E.2. Estimating the forward ESS

For the quantum mechanical particle in the symmetric double-well, we follow [7]. Ten Hamiltonian Markov Chains (HMCs) were run with 100k steps, 50 sub-steps and an overrelax frequency of 10, totaling in 1 million samples. We used 10k equilibrating steps. An overrelax step mirrors the sample around zero. Due to the symmetry of the Double-Well, both the original and the mirrored sample have the same probability. Thus the MC step would always be accepted.

E.3. Full results

Results for the forward and reverse ESS for $m_0=2.75$ can be found in table 3.

In figure 9 the forward and reverse ESS for $m_0\in\{2.75,3.03.25\}$ are shown. In particular, the top plot shows the results from table 3.

Table 3. Results of training a RealNVP for m_0: 2.75.

	d	ReinfPQ	ZPathPQ	PathPQ	RepQP	PathQP
FW ESS	8	0.99 ± .01	**1.00** ± .00	**1.00** ± .00	0.57 ± .20	**1.00** ± .00
	16	0.79 ± .18	**0.99** ± .00	**0.99** ± .00	0.79 ± .18	0.69 ± .18
	32	0.45 ± .18	**0.97** ± .01	**0.90** ± .05	0.06 ± .02	0.20 ± .17
	64	0.00 ± .00	**0.17** ± .15	**0.04** ± .03	0.00 ± .00	0.00 ± .00
Rev ESS	8	1.00 ± .00	**1.00** ± .00	**1.00** ± .00	**1.00** ± .00	**1.00** ± .00
	16	0.99 ± .00	**0.99** ± .00	**0.99** ± .00	0.99 ± .00	**0.99** ± .00
	32	0.95 ± .01	**0.98** ± .00	0.98 ± .00	0.85 ± .05	**0.98** ± .00
	64	0.48 ± .13	**0.90** ± .01	**0.69** ± .14	**0.60** ± .18	0.53 ± .13

Note: Best results within a statistical significance of p-value < 0.05, according to the Wilcoxon tests, are shown in bold.

[9] The reference refers to the PathPQ and ZPathPQ estimators for the loss of the variational encoder as IWAE-STL and RWS-DReG respectively.

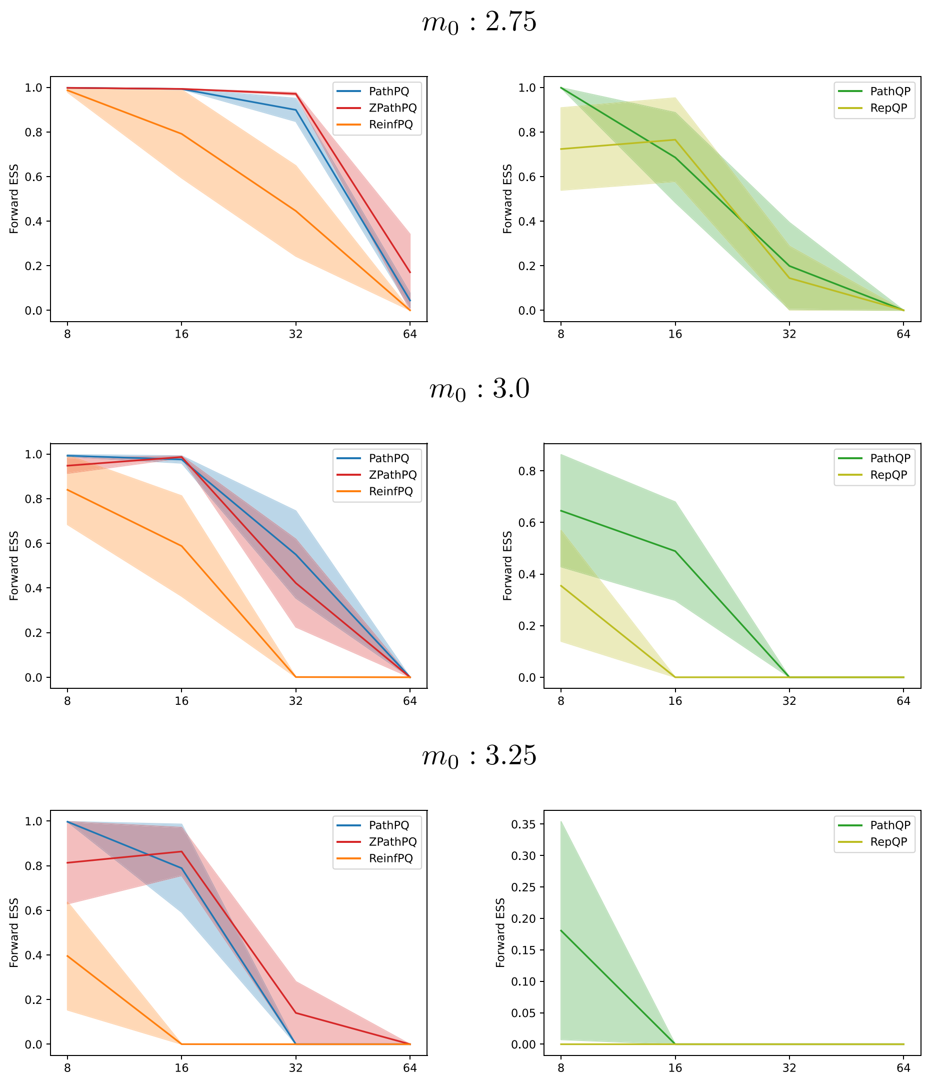

Figure 9. Forward ESS of trained RealNVP by varying over gradient estimators, dimensionality, and separation parameter m_0.

Appendix F. Sample Pytorch code for algorithm 1

```
def path_backward(flow, batch_size, action):
    """
    Parameters
    ----------
    flow : normalizing flow,
           has function forward & reverse,
           give back sample and log_q of the sample
           as well as sample_base
    batch_size : number of samples in batch
    action: function that computes action of sample
    """
    with torch.no_grad():
        z = flow.sample_base(batch_size)
        x1, _ = flow.forward(z)

    x2 = x1.requires_grad_()
    _, log_q = flow.reverse(x2)

    log_p = -action(x2)
    log_w_tilde = log_p - log_q
```

```
# The grad call deletes the computational graph from memory
# When using e.g. PathPQ, we use different grad_outputs
grad = torch.autograd.grad(outputs = log_w_tilde,
 grad_outputs = -torch.ones_like(log_w_tilde), inputs = x2)[0]
# Add to avoid rare instabilities
# grad = torch.where(grad! = grad, torch.zeros_like(grad), grad)

x3, _ = flow.forward(z)

(x3 * grad).mean().backward()
```

ORCID iD

Kim A Nicoli https://orcid.org/0000-0001-5933-1822

References

[1] Noé F, Olsson S, Köhler J and Hao W 2019 Boltzmann generators: sampling equilibrium states of many-body systems with deep learning *Science* **365** eaaw1147

[2] Hao W *et al* 2020 Stochastic normalizing flows *Advances in Neural Information Processing Systems (NeurIPS)* (arXiv:2002.06707)

[3] Albergo M S, Kanwar G and Shanahan P E 2019 Flow-based generative models for Markov chain Monte Carlo in lattice field theory *Phys. Rev.* D **100** 034515

[4] Kanwar G, Albergo M S, Boyda D, Cranmer K, Hackett D C, Racaniere Sebastien, Rezende D J and Shanahan P E 2020 Equivariant flow-based sampling for lattice gauge theory *Phys. Rev. Lett.* **125** 121601

[5] Boyda D, Kanwar G, Racanière S, Rezende D J, Albergo M S, Cranmer K, Hackett D C and Shanahan P E 2021 Sampling using SU(N) gauge equivariant flows *Phys. Rev.* D **103** 074504

[6] Dian W, Wang L and Zhang P 2019 Solving statistical mechanics using variational autoregressive networks *Phys. Rev. Lett.* **122** 080602

[7] Nicoli K A, Anders C J, Funcke L, Hartung T, Jansen K, Kessel P, Nakajima S and Stornati P 2021 Estimation of thermodynamic observables in lattice field theories with deep generative models *Phys. Rev. Lett.* **126** 032001

[8] Nicoli K A, Nakajima S, Strodthoff N, Samek W, Müller K-R and Kessel P 2020 Asymptotically unbiased estimation of physical observables with neural samplers *Phys. Rev.* E **101** 023304

[9] Nicoli K A, Kessel P, Strodthoff N, Samek W, Müller K-R and Nakajima S 2019 Comment on "Solving statistical mechanics using vans": introducing savant-vans enhanced by importance and mcmc sampling (arXiv:1903.11048)

[10] Müller T, McWilliams B, Rousselle F, Gross M and Novák J 2019 Neural importance sampling *ACM Trans. Graph.* **38** 145

[11] Hackett D C, Hsieh C-C, Albergo M S, Boyda D, Chen J-W, Chen K-F, Cranmer K, Kanwar G and Shanahan P E 2021 Flow-based sampling for multimodal distributions in lattice field theory (arXiv:2107.00734)

[12] Nicoli K A, Anders C J, Funcke L, Hartung T, Jansen K, Kessel P, Nakajima S and Stornati P 2021 Machine learning of thermodynamic observables in the presence of mode collapse *38th Int. Symp. on Lattice Field Theory* (https://doi.org/10.22323/1.396.0338)

[13] Roeder G, Yuhuai W and Duvenaud D 2017 Sticking the landing: simple, lower-variance gradient estimators for variational inference *Advances in Neural Information Processing Systems (NeurIPS)* (arXiv:1703.09194)

[14] Vaitl L, Nicoli K A, Nakajima S and Kessel P 2022 Path-gradient estimators for continuous normalizing flows *Int. Conf. on Machine Learning (ICML)* pp 21945–59 (arXiv:2206.09016)

[15] Tucker G, Lawson D, Shixiang G and Maddison C J 2019 Doubly reparameterized gradient estimators for Monte Carlo objectives *Int. Conf. on Learning Representations (ICLR)* (arXiv:1810.04152)

[16] Burda Y, Grosse R B and Salakhutdinov R 2016 Importance weighted autoencoders *Int. Conf. on Learning Representations (ICLR)* (arXiv:1509.00519)

[17] Bornschein J and Bengio Y 2015 Reweighted wake-sleep *Int. Conf. on Learning Representations (ICLR)* (arXiv:1406.2751)

[18] Nowozin S 2018 Debiasing Evidence Approximations: On Importance-weighted Autoencoders and Jackknife Variational Inference *Int. Conf. on Learning Representations (ICLR)*

[19] Williams R J 1992 Simple statistical gradient-following algorithms for connectionist reinforcement learning *Mach. Learn.* **8** 229–56

[20] Finke A and Thiery A H 2019 On importance-weighted autoencoders (arXiv:1907.10477)

[21] Bauer M and Mnih A 2021 Generalized doubly-reparameterized gradient estimators *Int. Conf. on Machine Learning (ICML)* pp 738–47 (arXiv:2101.11046)

[22] Geffner T and Domke J 2020 On the difficulty of unbiased alpha divergence minimization (arXiv:2010.09541)

[23] Geffner T and Domke J 2021 Empirical evaluation of biased methods for alpha divergence minimization (arXiv:2105.06587)

[24] Agrawal A, Sheldon D R, Domke J and Lin H-T 2020 Advances in black-box VI: normalizing flows, importance weighting and optimization *Advances in Neural Information Processing Systems (NeurIPS)* (arXiv:2006.10343)

[25] Kingma D P and Welling M 2014 Auto-encoding variational bayes *Int. Conf. on Learning Representations (ICLR)* (arXiv:1312.6114)

[26] Figurnov M, Mohamed S and Mnih A 2018 Implicit reparameterization gradients *Advances in Neural Information Processing Systems (NeurIPS)* (arXiv:1805.08498)

[27] Köhler J, Krämer A and Noé F 2021 Smooth normalizing flows *Advances in Neural Information Processing Systems (NeurIPS)* pp 2796–809 (arXiv:2110.00351)

[28] Ruiz F J R, Titsias M K and Blei D M 2016 The generalized reparameterization gradient *Advances in Neural Information Processing Systems (NeurIPS)* (arXiv:1610.02287)

[29] Jankowiak M and Obermeyer F 2018 Pathwise derivatives beyond the reparameterization trick *Int. Conf. on Machine Learning (ICML)* pp 2235–44 (arXiv:1806.01851)

[30] Wan N, Dapeng L and Hovakimyan N 2020 F-divergence variational inference *Advances in Neural Information Processing Systems (NeurIPS)* pp 17370–9 (arXiv:2009.13093)

[31] Naesseth C, Ruiz F, Linderman S and Blei D 2017 Reparameterization gradients through acceptance-rejection sampling algorithms *Artificial Intelligence and Statistics* (arXiv:1610.05683)

[32] Mnih A and Gregor K 2014 Neural variational inference and learning in belief networks *Int. Conf. on Machine Learning (ICML)* (arXiv:1402.0030)

[33] Richter L, Boustati A, Nüsken N, Ruiz F and Akyildiz O D 2020 VarGrad: a low-variance gradient estimator for variational inference *Advances in Neural Information Processing Systems (NeurIPS)* pp 13481–92 (arXiv:2010.10436)

[34] Salimans T and Knowles D A 2014 On using control variates with stochastic approximation for variational bayes and its connection to stochastic linear regression (arXiv:1401.1022)

[35] Hesterberg T C 1988 Advances in importance sampling

[36] Dhaka A K, Catalina A, Welandawe M, Andersen M R, Huggins J H and Vehtari A 2021 Challenges and opportunities in high-dimensional variational inference *CoRR* (arXiv:2103.01085)

[37] Owen A B 2013 Monte Carlo theory, methods and examples (available at: https://artowen.su.domains/mc/)

[38] Bugallo M F, Elvira V, Martino L, Luengo D, Miguez J and Djuric P M 2017 Adaptive importance sampling: the past, the present and the future *IEEE Signal Process. Mag.* **34** 60–79

[39] Neal R M 2001 Annealed importance sampling *Stat. Comput.* **11** 125–39

[40] Midgley L I, Stimper V, Simm G N C, Schölkopf B and Hernández-Lobato J M 2022 Flow annealed importance sampling bootstrap (arXiv:2208.01893)

[41] Laszkiewicz M, Lederer J and Fischer A 2022 Marginal tail-adaptive normalizing flows *Int. Conf. on Machine Learning (ICML)* pp 12020–48 (arXiv:2206.10311)

[42] Jaini P, Kobyzev I, Yaoliang Y and Brubaker M 2020 Tails of lipschitz triangular flows *Int. Conf. on Machine Learning (ICML)* pp 4673–81 (arXiv:1907.04481)

[43] Aaron van den O *et al* 2018 Parallel WaveNet: fast high-fidelity speech synthesis *Int. Conf. on Machine Learning (ICML)* pp 3918–26 (arXiv:1711.10433)

[44] Casella G and Berger R L 2021 *Statistical Inference* (Boston, MA: Cengage Learning)

[45] Small C G 2010 *Expansions and Asymptotics for Statistics* (Boca Raton, FL: CRC Press)